1. A method of query pattern matching, comprising:

(a) generating a list of potential ancestors and a list of potential descendants;

(b) sorting the list of potential ancestors and the list of potential descendants in order of a first attribute in a database;

(c) skipping over unmatchable nodes in the list of potential descendants;

(d) determining whether a second attribute of a current node in the potential descendant list is less than a second attribute of a current node in the potential ancestor list;

(e) determining, based upon a result from (d), whether a first attribute of the current node of the potential ancestor list is less than a first attribute of the current node of the potential descendant list, a second attribute of the current node of the potential descendant list is less than a second attrubute of the current node of the potential ancestor list, and a level number of the current node of the potential descendant list is equal to a level number plus one of the current node of the potential ancestor list; and

(f) appending to an output join list, based upon a result from (e), a node pair comprising the current node of the potential ancestor list and the current node of the potential descendant list.

2. The method according to claim 1, wherein the first attribute corresponds to a start position.

3. The method according to claim 1, wherein the second attribute corresponds to an end position.

4. The method according to claim 1, further including matching the query pattern against an XML database.

5. The method according to claim 1, further including sorting the output join list in ancestor/parent order.

6. A method of query pattern matching, comprising:

(a) generating a list of potential ancestors and a list of potential descendants;

(b) sorting the list of potential ancestors and the list of potential descendants in order of a start position attribute in a database;

(c) skipping over unmatchable nodes in the list of potential ancestors;

(d) determining whether a start position of a current node in the potential ancestor list is less than a start position of a current node in the potential descendant list;

(e) determining, based upon a result from (d), whether a start position of the current node of the potential ancestor list is less than a start position of the current node of the list of potential descendants, an end position of the current node of the potential descendant list is less than an end position of the current node of the potential ancestor list, and a level number of the current node of the potential descendant list is equal to a level number plus one of the current node of the potential ancestor list; and

(f)) appending to an output join list, based upon a result from (e), a node pair comprising the current node of the potential ancestor list and the current node of the potential descendant list.

7. The method according to claim 6, further including matching the query pattern in an XML document.

8. The method according to claim 6, further including sorting the output join list in descendant/child order.

9. A method of query pattern matching, comprising:

(a) generating a list of potential ancestors and a list of potential descendants;

(b) sorting the list of potential ancestors and the list of potential descendants in order of a start position attribute in a database;

(c) determining whether input lists are not empty and whether a stack is not empty;

(d) determining, based upon a result from (c), whether a start position of a current node of the list of potential ancestors is greater than an end position of a node on a top of the stack and a start position of a current node of the list of potential descendants is greater than the end position of the node on the top of the stack;

(e) popping the stack for a first result from (d) and for a second result from (d) determining whether the start position of the current node of the list of potential ancestors is less than the start position of the current node of the list of potential descendents;

(f) pushing the current node of the list of potential ancestors onto the stack and looking to a next node of the list of potential ancestors for a first result from (e) and appending matches to an output list and looking to a next node in the list of potential descendants for a second result from (e).

10. The method according to claim 9, further including matching a pattern in an XML document.

11. The method according to claim 9, further including sorting output in descendant order.

12. The method according to claim 11, further including making a single pass though the list of potential ancestors.

13. The method according to claim 11, further including making a single pass through the list of potential descendants.

14. The method according to claim 9, further including sorting the output list based upon one or more of document ID, start position of node from the list of potential ancestors, and start position of node from the list of potential descendants.

15. A method of query pattern matching, comprising:

(a) generating a list of potential ancestors and a list of potential descendants;

(b) sorting the list of potential ancestors and the list of potential descendants in order of a start position attribute in a database;

(c) determining whether input lists are not empty and whether a stack is not empty;

(d) determining, based upon a result from (c), whether a start position of a current node of the list of potential ancestors is greater than an end position of a node on a top of the stack and a start position of a current node of the list of potential descendants is greater than the end position of the node on the top of the stack;

(e) for a first result from (d) popping the stack, determining whether the stack is empty and if the stack is empty merging a tuple self and inherit lists into an inherit list associated with a top of the stack and if the stack is not empty output the tuple self and inherit list, and for a second result from (d) determining whether the start position of the current node of the list of potential ancestors is less than the start position of the current node of the list of potential descendents;

(f) pushing the current node of the list of potential ancestors onto the stack and looking to a next node of the list of potential ancestors for a first result from (e) and appending matches to an output list and looking to a next node in the list of potential descendants for a second result from (e).

16. The method according to claim 15, further including matching a pattern in an XML database.

17. The method according to claim 15, further including making a single pass though the list of potential ancestors.

18. The method according to claim 17, further including making a single pass through the list of potential descendants.

19. The method according to claim 15, further including sorting the output list based upon one or more of document ID, start position of node from the list of potential ancestors, and start position of node from the list of potential descendants.